The Web

IN4325 – Information Retrieval

Today

- The Web
- HITS & PageRank
- Sentiment Analysis
  - Naïve Bayes classifier
- Spam
  - Decision tree classifier
- The Social Web
  - Personalized search

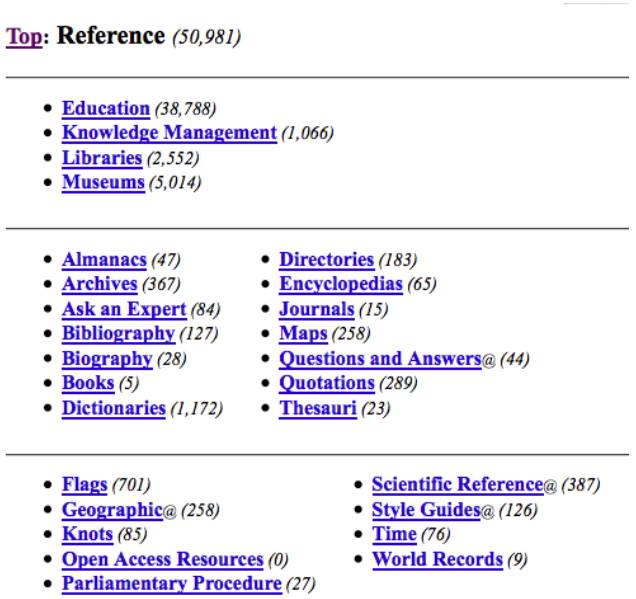
next Monday
Prof. Arjen de Vries on entities

next Wednesday
search engine clicks
(ranking & advertising)

The Web graph

The Web

Open Directory Project:
5,007,664 sites
94,441 editors
over 1,010,258 categories



Top: Reference (50,981)

- Education (38,788)
- Knowledge Management (1,066)
- Libraries (2,552)
- Museums (5,014)

- Almanacs (47)
- Archives (367)
- Ask an Expert (84)
- Bibliography (127)
- Biography (28)
- Books (5)
- Dictionaries (1,172)
- Directories (183)
- Encyclopedias (65)
- Journals (15)
- Maps (258)
- Questions and Answers@ (44)
- Quotations (289)
- Thesauri (23)

- Flags (701)
- Geographic@ (258)
- Knots (85)
- Open Access Resources (0)
- Parliamentary Procedure (27)
- Scientific Reference@ (387)
- Style Guides@ (126)
- Time (76)
- World Records (9)

- Vannevar Bush envisioned hypertext in the 1940's
- First hypertext systems were created in the 1970's

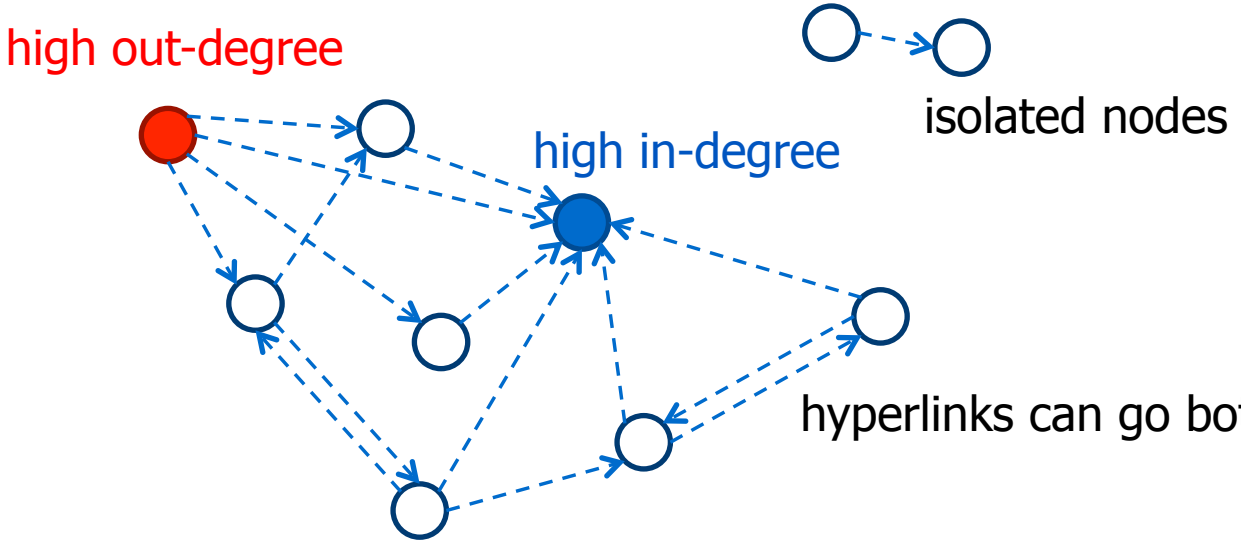
- The World Wide Web was formed in the early 1990's
  - Creator: Tim Berners-Lee
  - make documents easily available to anyone on the Internet (*Web pages*)
  - Easy access to such Web pages using a *browser*

- Early Web years
  - Full-text search engines (Altavista, Excite and Infoseek) vs.
  - Taxonomies populated with pages in categories (**ODP**, Yahoo! Directory)

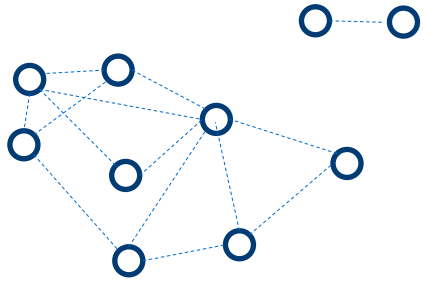
The Web

- Today the size of the Web makes it impossible to find anything without the help of search engines
  - Estimating the size of the Web is a research area by itself (by comparing the overlap between Web search engines output)
  - In 1998 (Google): 24 million pages (at that time: academic engine)
  - In 1999 (Altavista): >200 million pages
  - In 2005: *indexed* Web estimated to be at 11.5 billion pages [7]
- Users view the Web through the lense of the search engine
- Pages not indexed (or ranked at low positions) by search engines are unlikely to be found by users

The World Wide Web



For analysis purposes, sometimes:



Research questions

Why do people link?
How connected is the web graph?
What properties does the graph have?
How can we exploit those properties?

Web graph

- **nodes**: web pages, web sites, domains (dependent on abstraction level
- **directed edges**: hyperlinks

Graph structure in the Web

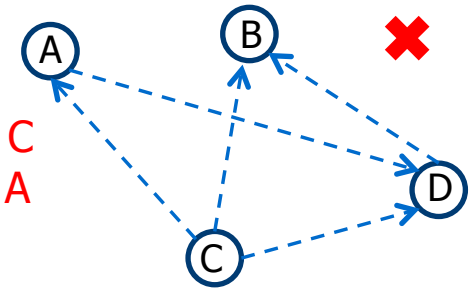
Broder et al., 1999 [4] >2000 citations

- Insights important for
  - The design of crawling strategies
  - Understanding the sociology of content creation on the Web
  - Analyzing the behaviour of algorithms that rely on link information (e.g. HITS, PageRank)
  - Predicting the evolution of web structures
  - Predicting the emergence of new phenomena in the Web graph
- Data: Altavista crawl from 1999 with 200M pages and 1.5 billion links
- In/out-degrees follow power laws
  - Probability that a node has in/out-degree i is proportional to $\frac{1}{i^x}, x > 1$

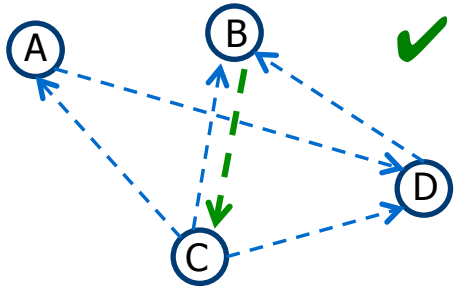
Graph theory: connected components

A detour

- Strongly connected component (SCC): directed graph with a path from each node to every other node



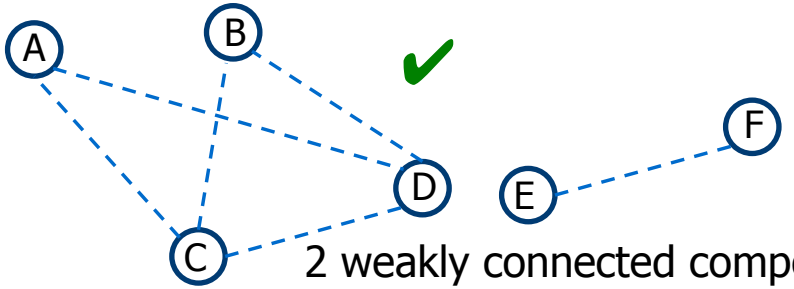
A cannot reach C
B cannot reach A
.....



$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

adjacency matrix

- Weakly connected component (WCC): directed graph with a path in the underlying undirected graph from each node to every other node



2 weakly connected components

Graph theory: connected components

A detour

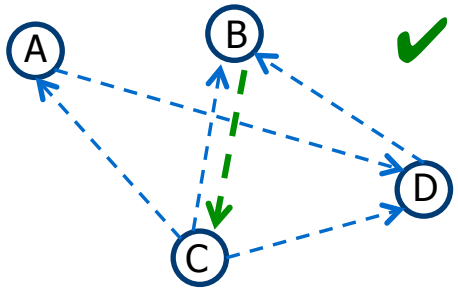
- Strongly connected component (SCC): directed graph with a path from each node to every other node

$G = (V, E)$

$V = \{A, B, C, D\}$

$E = \{(A,D),(B,C),(C,A),(C,B),(C,D),(D,B)\}$

$d(A,B) = 2,\ d(C,B) = 1,\ d(A,C) = 3$



$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

adjacency matrix

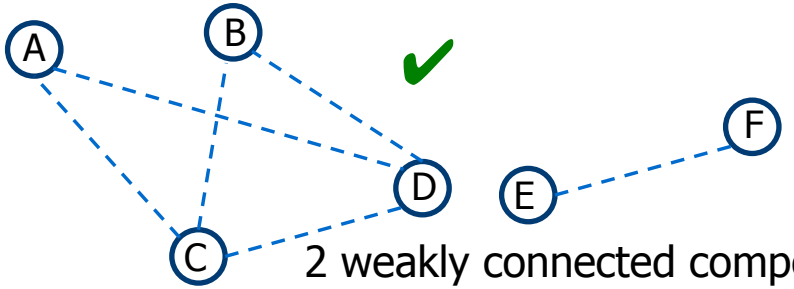
- Weakly connected component (WCC): directed graph with a path in the underlying undirected graph from each node to every other node

$G = (V, E)$

$V = \{A, B, C, D\}$

$E = \{\{A,C\},\{A,D\},\{B,C\},\{B,D\},\{C,D\}\}$

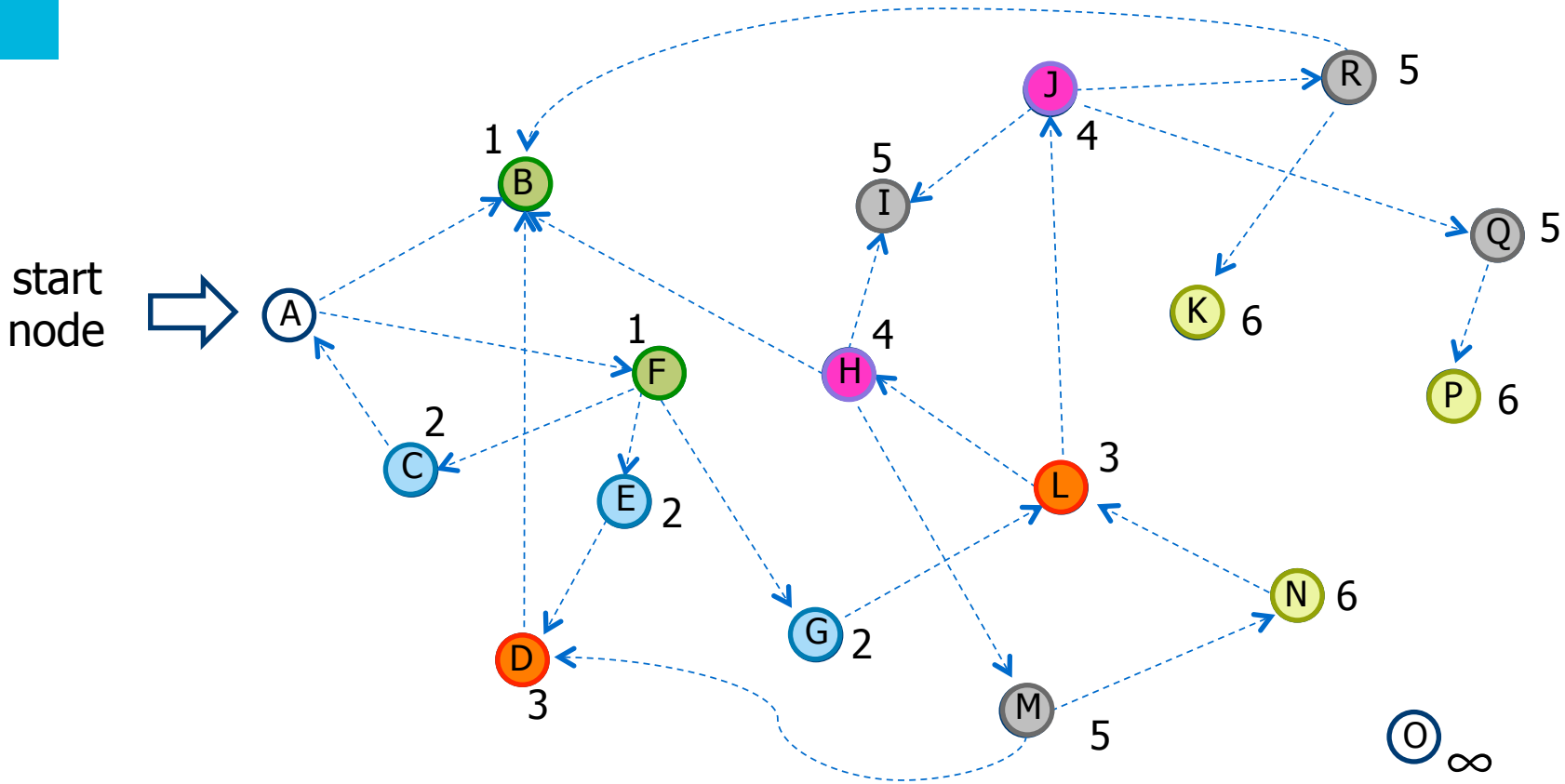
$d(A,B) = 2,\ d(C,B) = 1,\ d(A,C) = 1$



2 weakly connected components

Graph theory: breadth-first search

A detour

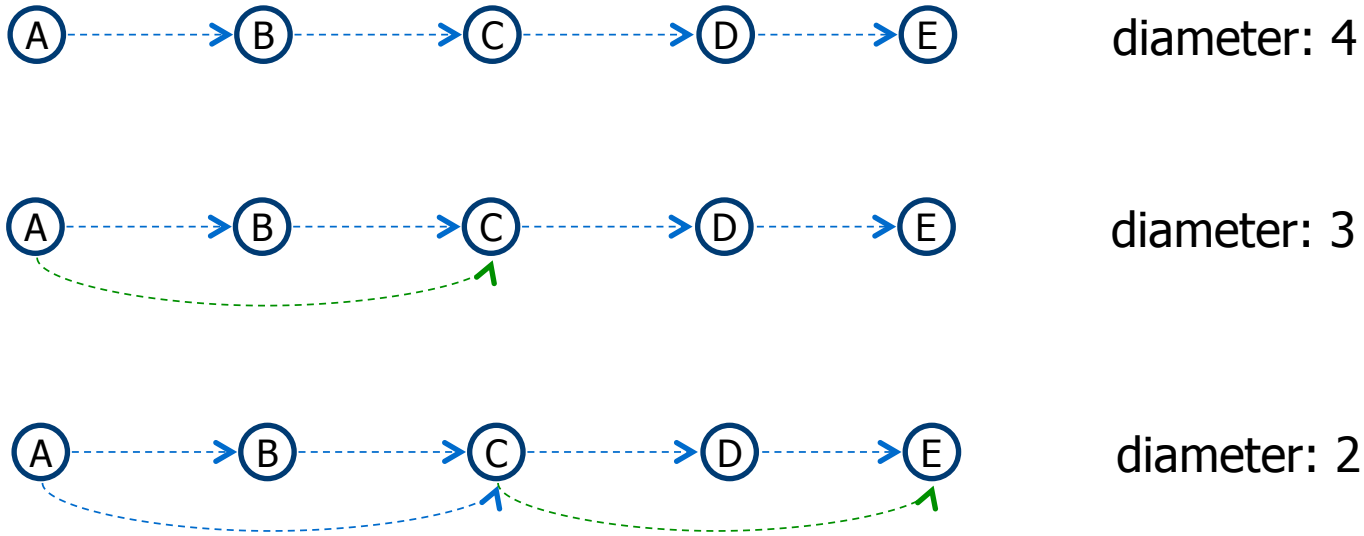


Graph theory: diameter

A detour

- Graph diameter: longest shortest path in the graph

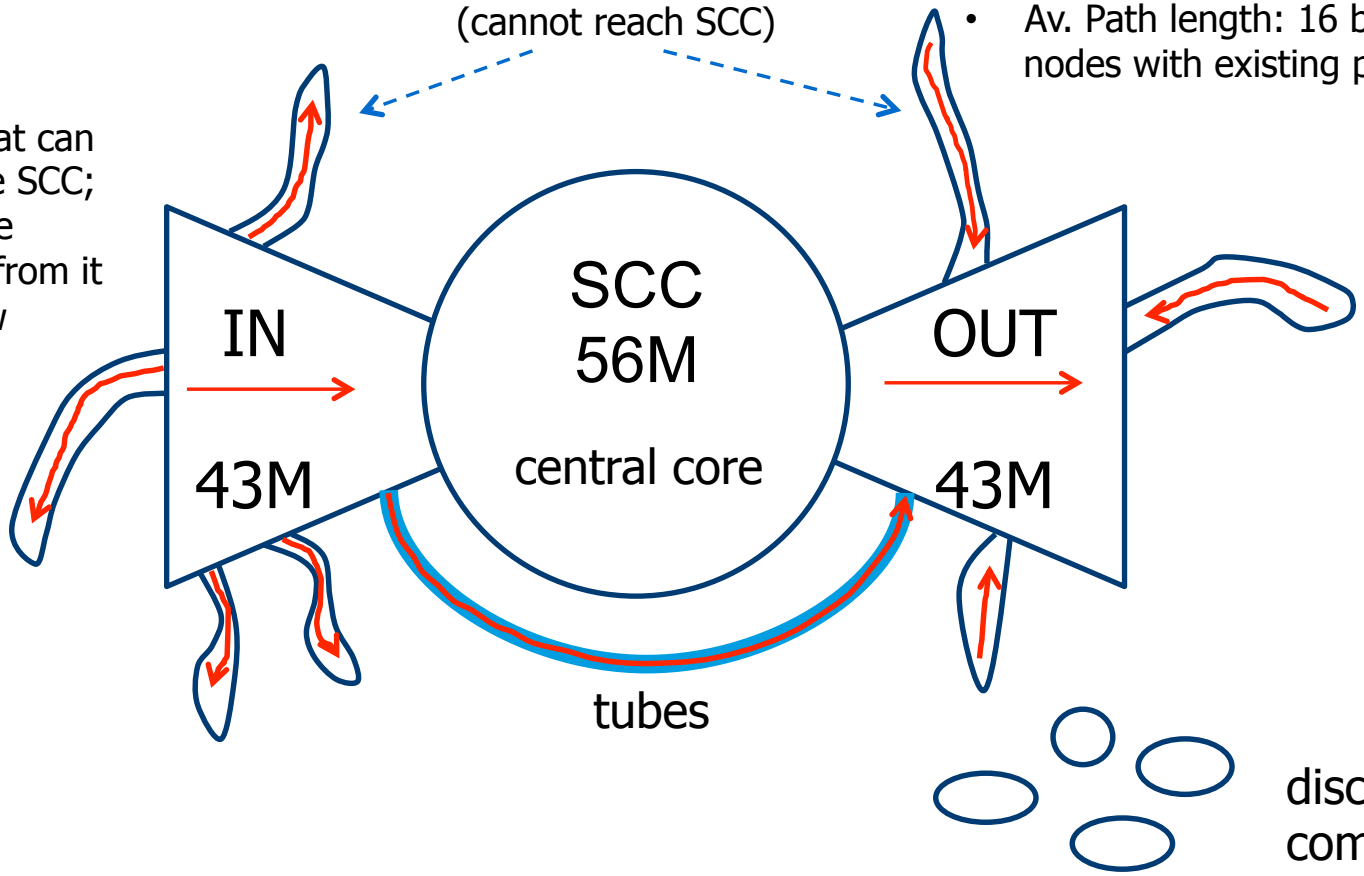
$$\max_{x,y \in V} d(x,y)$$



Graph structure in the Web

Broder et al., 1999 [4]

- ~200M nodes in total
- >90% in a single WCC
- Av. connected distance SCC: 28
- Av. connected distance graph: >500
- Av. Path length: 16 between any two nodes with existing path



Source: [4]

Graph structure in the Web

Broder et al., 1999 [4]

- "In a sense the web is much like a complicated organism, in which the local structure at a microscopic scale looks very regular like a biological cell, but the global structure exhibits interesting morphological structure (body and limbs) that are not obviously evident in the local structure."

Evolution of the Web

Fetterly et al., 2003 [8]

- How fast does the Web change?
  - Important to build effective crawlers (what portions of the index should be updated?
- Setup: weekly crawl of 150M pages once a week for 11 weeks (2002-2003)

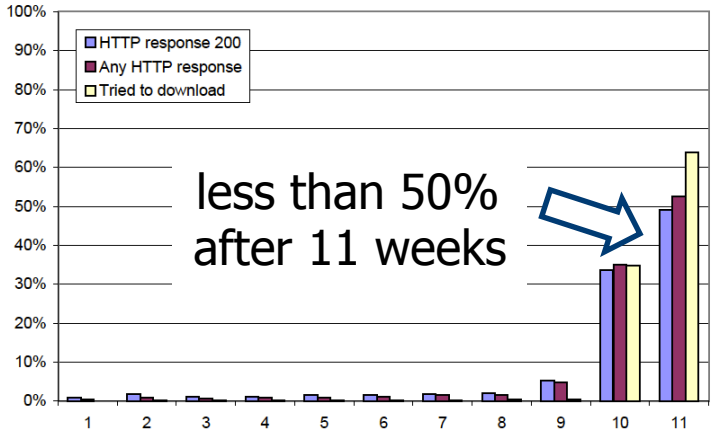


Figure 1. Number of successful, completed, and attempted downloads per URL.

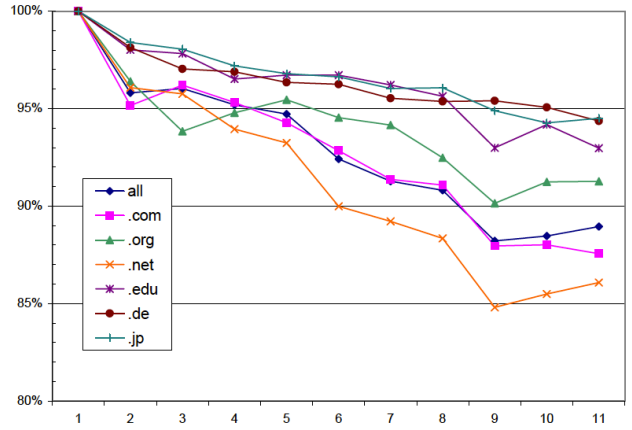
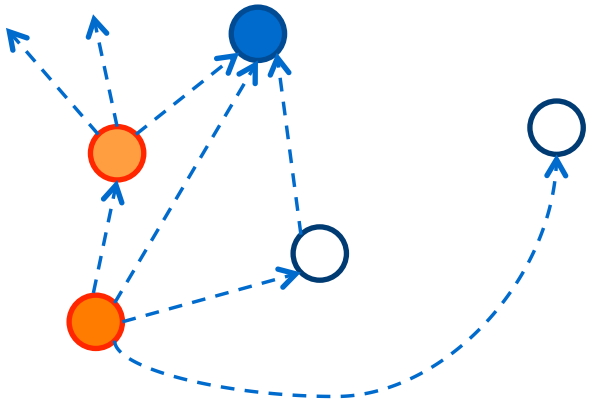


Figure 5. Distribution of successful downloads over crawl generations, broken down by selected top-level domains.

Source: [7]

HITS

Kleinberg, 1998 [12]



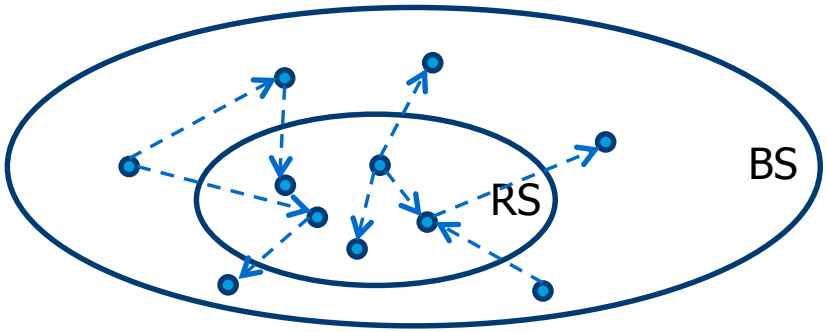
- **H**yperlink-**I**nduced **T**opic **S**earch

- Intuition: two broad types of useful pages for ad hoc search queries
  - **Authoritative** pages: pages containing a lot of relevant information about the search topic
    - Wikipedia pages, etc.
    - High weight $a(p)$
  - **Hub** pages: pages containing a large number of useful hyperlinks pointing to pages with relevant content
    - Yahoo! Portal, Open Directory Project, etc.
    - High weight $h(p)$

a page pointed to by many hubs

a page pointed to by many authorities

HITS



① Root set (RS): retrieve the top 200 results for a given keyword query

② Base set (BS): expand RS by including all* pages that link to pages in RS or are linked-to by pages in RS

③ Clean hyperlink structure (link removal between pages belonging to the same web site)

④ Initialize all hub/authority weights to 1

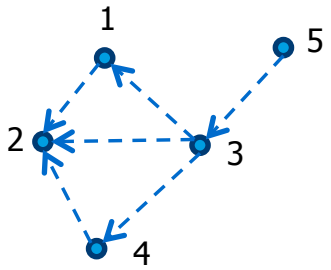
⑤ Iteratively update hub/authority weights (& normalize)

$$a(p) = \sum_{q \to p} h(q) \qquad h(p) = \sum_{p \to q} a(q)$$

authority weight increased if good hubs point to p

hub weight of p increased if it points to good authorities

HITS



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$a(p) = \sum_{q \to p} h(q)$$

$$h(p) = \sum_{p \to q} a(q)$$

initialize :

$\vec{a} = (a(1),...,a(5)) = (1,1,1,1,1)$

first round :

fixed!

$a(1) = h(3) = 1 \propto \frac{1}{6}$

$a(2) = h(1) + h(3) + h(4) = 1 + 1 + 1 \propto \frac{1}{2}$

$a(3) = h(5) = 1 \propto \frac{1}{6}$

$a(4) = h(3) = 1 \propto \frac{1}{6}$

$a(5) = 0$

initialize :

$\vec{h} = (h(1),...,h(5)) = (1,1,1,1,1)$

first round :

$h(1) = a(2) = 1 \propto \frac{1}{6}$

$h(2) = 0$

$h(3) = a(1) + a(2) + a(4) = 3 \propto \frac{1}{2}$

$h(4) = a(2) = 1 \propto \frac{1}{6}$

$h(5) = a(3) = 1 \propto \frac{1}{6}$

normalization after each round (sum to 1)

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 3 \\ 1 \\ 1 \end{pmatrix}$$

$A \qquad \times \vec{a}_{init} = \vec{h}_1$

$$\vec{h}_m = A\vec{a}_{m-1}$$

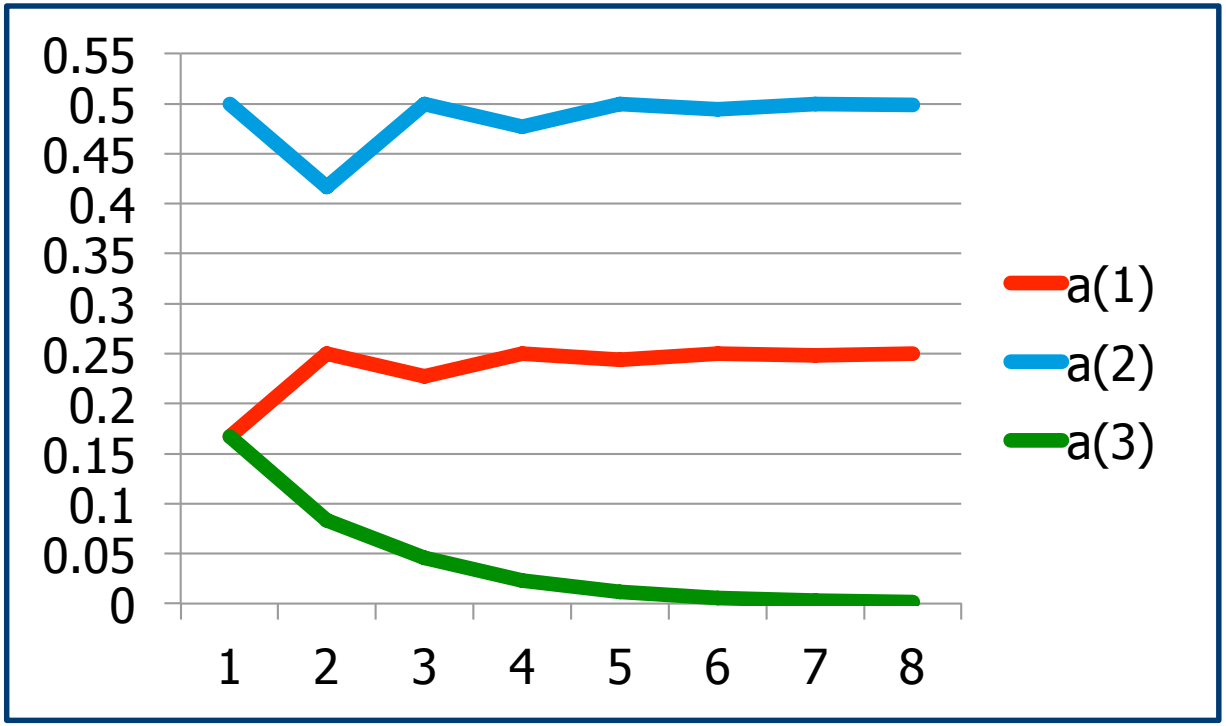
$$\vec{a}_m = A^T \vec{h}_{m-1}$$

$$\vec{h}_m = (A \times A^T)\vec{h}_{m-2}$$

$$\vec{a}_m = (A^T \times A)\vec{a}_{m-2}$$

HITS

Our toy example: weights across 8 iterations



HITS output: a list of top-scoring authority/hub pages for the given query

PageRank

Page et al., 1998 [5]

>4700 citations

- A **topic independent** approach to page importance
  - Computed once per crawl
- Every document of the corpus is assigned an importance score
  - In search: re-rank (or filter) results with a low PageRank score
- Simple idea: number of in-links ⇨ importance
  - Page p_1 has 10 in-links and one of those is from yahoo.com, page p_2 has 50 in-links from obscure pages
- PageRank takes the importance of the page where the link originates into account

"To test the utility of PageRank for search, we built a web search engine called Google."

PageRank

Page et al., 1998 [5]

- Idea: if page p_x links to page p_y, then the creator of p_x implicitly transfers some importance to page p_y
  - yahoo.com is an important page, many pages point to it
  - pages linked to from yahoo.com are also likely to be important
- Each page distributes "importance" through its outlinks
- Simple PageRank (iteratively):

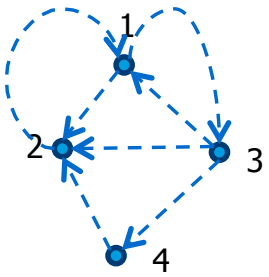
$$PageRank_{i+1}(v) = \sum_{u \to v} \frac{PageRank_i(u)}{N_u}$$

all nodes linking to v — out-degree of node u

A page with many out-links has little influence on one particular page.

PageRank

Simplified formula



$$W = \begin{pmatrix} 0 & 1 & 1/3 & 0 \\ 1/2 & 0 & 1/3 & 1 \\ 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \end{pmatrix}$$

outlinks of node 1

$$w_{ij} : \frac{link\ j \rightarrow i}{outlinks_j}$$

initialize PageRank vector $\vec{R}$

$$\vec{R} = (R(1),...,R(4)) = (0.25,0.25,0.25,0.25)$$

$$PageRank_i = W \times PageRank_{i-1}$$

$$W^1 \times \vec{R}' = \begin{pmatrix} 0.33 \\ 0.46 \\ 0.13 \\ 0.08 \end{pmatrix}$$

$$W^2 \times \vec{R}' = \begin{pmatrix} 0.50 \\ 0.29 \\ 0.17 \\ 0.04 \end{pmatrix}$$

$$W^3 \times \vec{R}' = \begin{pmatrix} 0.35 \\ 0.35 \\ 0.25 \\ 0.06 \end{pmatrix}$$

$$W^{16} \times \vec{R}' = \begin{pmatrix} 0.40 \\ 0.33 \\ 0.20 \\ 0.07 \end{pmatrix}$$

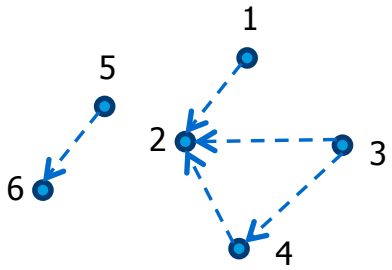
$$W^{17} \times \vec{R}' = \begin{pmatrix} 0.40 \\ 0.34 \\ 0.20 \\ 0.07 \end{pmatrix}$$

PageRank vector converges eventually

Random surfer model

- probability that a random surfer starts at a random page and ends at page p_x
- a random surfer at page p_x with three outlinks randomly picks one (thus 1/3)

PageRank



$$W = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1/2 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \end{pmatrix}$$

Reality

initialize PageRank vector $\vec{R}$

$$\vec{R} = (R(1),...,R(4)) = (0.25, 0.25, 0.25, 0.25)$$

disconnected components

nodes without outgoing edges lead to problems (**rank sink**)

$$W^1 \times \vec{R}' = \begin{pmatrix} 0.00 \\ 0.63 \\ 0.00 \\ 0.13 \end{pmatrix}$$

$$W^2 \times \vec{R}' = \begin{pmatrix} 0.00 \\ 0.13 \\ 0.00 \\ 0.00 \end{pmatrix}$$

$$W^3 \times \vec{R}' = \begin{pmatrix} 0.00 \\ 0.00 \\ 0.00 \\ 0.00 \end{pmatrix}$$

Include a decay ("damping") factor

$$PageRank_{i+1}(v) = p \sum_{u \to v} \frac{PageRank_i(u)}{N_u} + (1-p)$$

probability that the random surfer "teleports" and not uses the outlinks

PageRank applications

- Search
  - Re-rank the top retrieved documents of a content retrieval technique according to the pages' PageRank score
  - Filter out pages with low PageRank scores
- Personalized PageRank
  - Instead of random teleporting, bias the teleport locations
- PageRank as future inlink count predictor
  - Re-order crawling list accordingly (crawl better pages first)

Opinion mining

Opinion mining

- Opinion mining is the automatic extraction of subjective information from documents

- Applications
  - Automatic review aggregation websites (what do people think about X?)
  - Recognition of user rating errors: review vs. rating polarity
  - Flame detection & detection of sensible content (e.g. accidents reports) to avoid showing inappropriate ads
  - Information extraction: discard information from subjective sentences
  - Summarization: present summaries with diverse points of view

Intuitive approach: list of strongly opinionated terms

Pang et al., 2002 [13]

- Idea: manually create a list of positive and negative terms and **classify** documents according to how frequent positive and negative terms occur

- Assessor 1 (for movie reviews)
  - positive: dazzling, brilliant, phenomenal, excellent, fantastic
  - negative: suck, terrible, awful, unwatchatble, hideous
  - Accuracy~58%, ties~75%

- Posthoc analysis
  - Positive: love, wonderful, best, great, superb, still beautiful
  - Negative: bad, worst, stupid, waste, boring, ?, !
  - Accuracy~69%, ties~16%

		gold standard true	gold standard false
algorithm	true	true positive (TP)	false positive (FP)
algorithm	false	false negative (FN)	true negative (TN)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

if the test data contains the same number of true/false items, a random "classifier" on av. achieves 50% accuracy

Machine learning for opinion mining [12]

Pang et al., 2002 [13]

- Predefined set of features $\{f_1, f_2, \ldots, f_m\}$ that are related to the classification task at hand
  - Unigram: *still*
  - Bigram: *really stinks*
  - POS tags
  - #exclamation marks
  - #question marks
  - #☺
  - #☹
  - .....

number of times f_i occurs in D

$$D = \begin{pmatrix} n_1(D) \\ n_2(D) \\ n_3(D) \\ \ldots \\ n_m(D) \end{pmatrix}$$

document as a vector
m dimensions=*#features*

Finding good features is hard!

Naïve Bayes

- A review is assigned the class $c' = \arg\max_c P(c \mid D)$
- Bayes' rule

$$P(c \mid D) = \frac{P(c)P(D \mid c)}{P(D)}$$

- Features are assumed to be conditionally independent given D's class

$$P(c \mid D) = \frac{P(c)\prod_{i=1}^{m} P(f_i \mid c)^{n_i(D)}}{P(D)}$$

Class prior: used when one class is known to occur more frequently than the other (e.g. spam emails vs. legitimite emails)

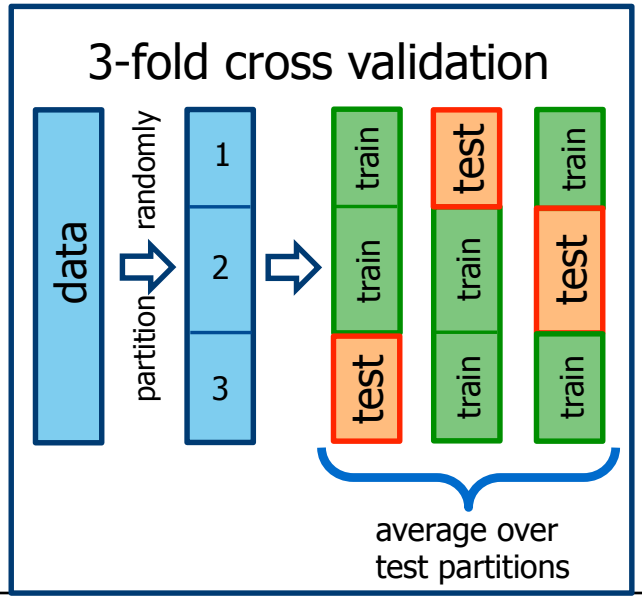
add-one smoothing (to avoid zero prob.)

Results

Pang et al., 2002 [13]

- Dataset: 700 positive and 700 negative reviews from IMDB
- Negation tagging: addition of *NOT_* to every word between a negation indicator and a punctuation mark
- 3-fold cross validation

	#features	freq. or pres.	accuracy
unigrams	16165	freq.	78.7
unigrams	16165	pres.	81.0
Unigrams+bigrams	32330	pres.	80.6
Unigrams+POS	16695	pres.	**81.5**
adjectives	2633	pres.	77.0



Sentiment anlysis challenges

- Sentiment polarity
  - The movie is not bad.
- Modeling sequential information
  - A is better than B. vs. B is better than A.
- Subtle sentiments
  - The polar express seems overly concerned with aping real life instead of creating its own universe.
- Distinction between opinions and facts
  - I love this movie. Vs. This is a love story.
- Documents often contain positive and negative opinions
  - The camera in the phone is great, but the radio is rubbish.

- Even more: cross-language mining, microblogs, emotions

Spam

Spam detection

- Spam: unsolicited (and possibly commercial) bulk messages
- Types of spam
  - Email
  - Instant messaging (spim)
  - Internet telephony (spit)
  - Mobile phone
  - Web

  Motivation: monetary

Web spam

- Spammers cannot send pages directly to the user, need "cooperation" of search engines (spamdexing)
- Users tend to click only on the top ranked results of search engine listings, thus spammers need to create pages that score highly
- Grey area between ethical Search Engine Optimization (SEO) and unethical spam
- Adversarial relationship between web site administrator and search engine administrator

Formally ....

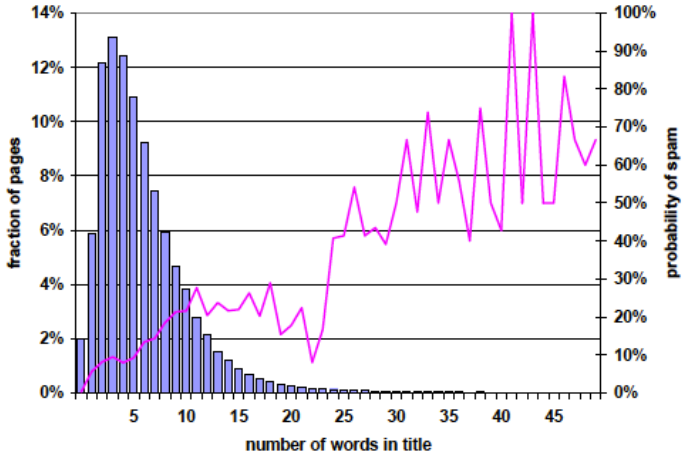
- Spamming [14]
  - "Any deliberate human action that is meant to trigger an unjustifiably favorable relevance or importance for some web page, considering the page's true value"
- Spamming [15]
  - "A web page created for the sole purpose of attracting search engine referrals (to this page or some other target page)."

Web spam classification

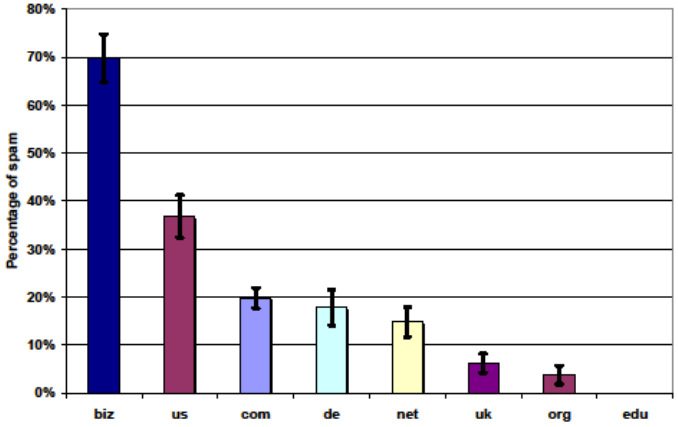
- Content spam: making changes to a web page's content [14, 9]
  - Dumping
  - Weaving
  - Phrase stitching
  - Keyword stuffing
  - …
- Topological spam: spamming with the help of link farms
- Cloaking [16]: "… a hiding technique used by some Web servers to deliver one page to a search engine for indexing while serving an entirely different page to users browsing the site."

Content spam

- Can be automatically detected with high accuracy [9]
- Features
  - Number of words in the page
  - Number of words in the page title
  - Average lenth of the words
  - Amount of anchor text
  - Fraction of the N most popular words in the page
  - Fraction of visible content
  - Compression ratio
  - …



num. words/title



spam%/domain

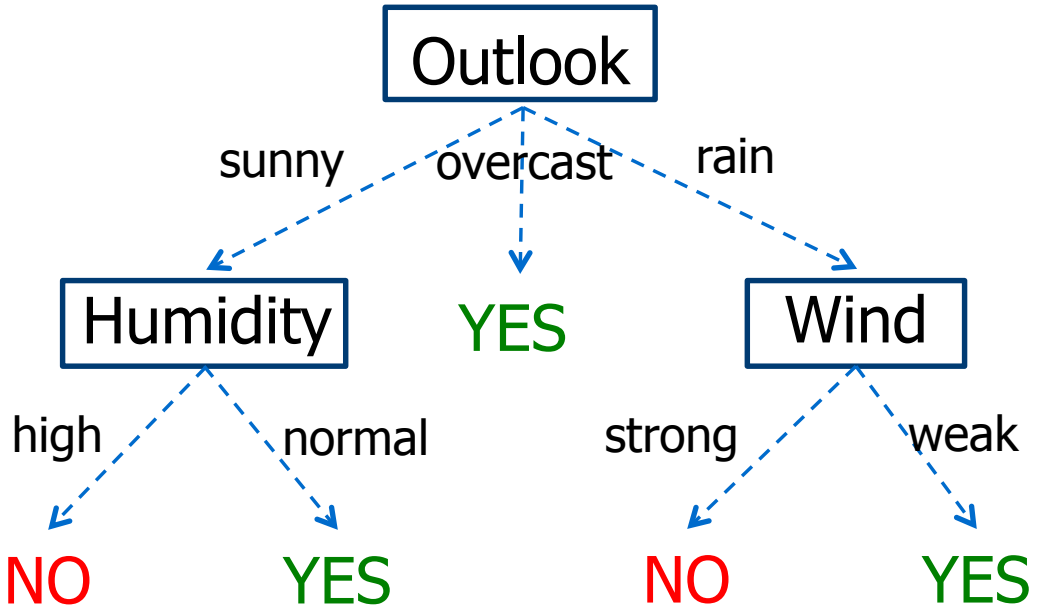
Source: [9]

Decision trees

A detour

- Widely used machine learning approach
- Learned function represented in a decision tree
- Human readable
- Robust to noise
- Can also be represented as if/then rules

$PlayTennis = \{YES, NO\}$



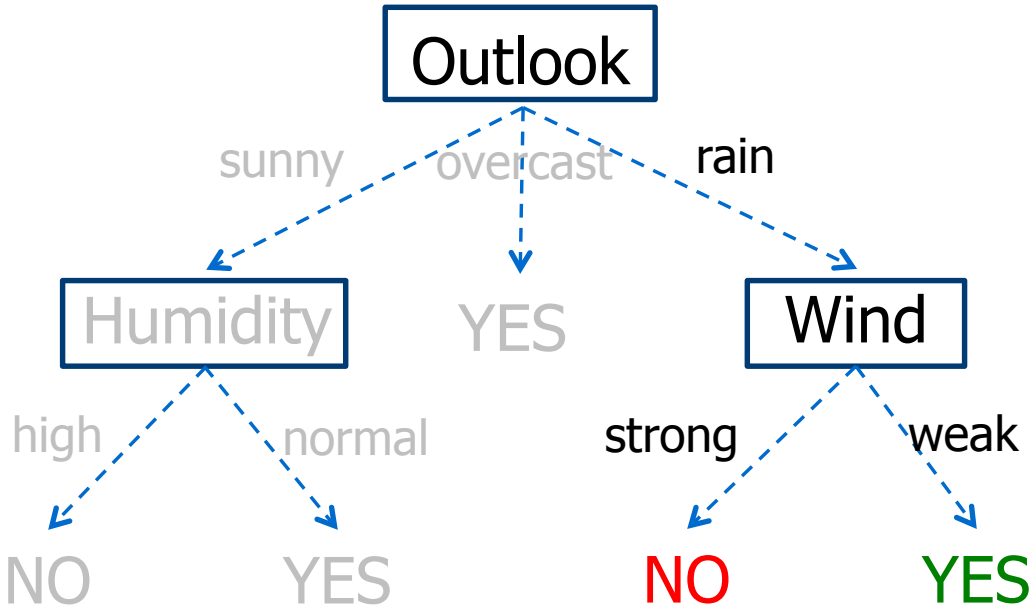
Decision tree overview taken from:
Machine Learning, Tom Mitchell, McGraw Hill, 1997.

Decision trees

A detour

- Widely used machine learning approach
- Learned function represented in a decision tree
- Human readable
- Robust to noise
- Can also be represented as if/then rules

$$PlayTennis=\{YES,NO\}$$



$$I=\{outlook_{rain},\ wind_{strong},\ humidity_{high}\}$$

Decision trees: appropriate when ...

A detour

- Instances are represented by attribute-value pairs
  - Either discrete (e.g. outlook={sunny,overcast,rain}) or real-valued
- Target function has discrete output values (e.g. *PlayTennis={YES,NO}*)
- The training data may contain errors
- The training data may contain missing attribute values

Decision trees

A detour

No back-tracking

- General idea: top-down greedy built-up of the tree

① Which attribute should be the tree's root/new node?
  - Teach each attribute alone, how well does it classify the training examples? Pick the best one.

② Child node created for each possible value of this attribute

③ Sort training instances accordingly

④ Go to step (1), repeat until all training instances classified

Decision trees: what is the 'best' attribute?

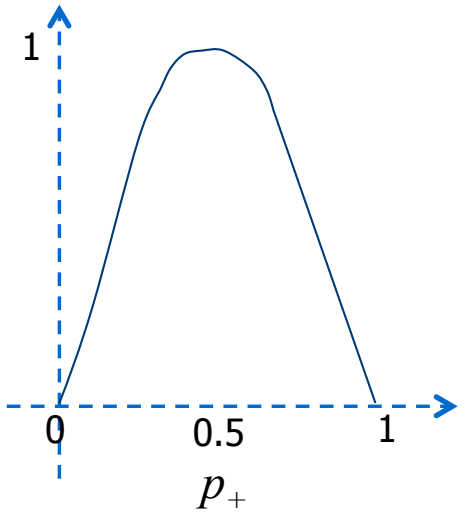
A detour

define:
$0 \log 0 = 0$

- $Entropy(S) = \sum_{i=1}^{r} -p_i \log_2 p_i$
- Example: 2 classes, 14 instances, 9+ and 5-

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

$$= -\frac{9}{14}\log_2 \frac{9}{14} - \frac{5}{14}\log_2 \frac{5}{14} = 0.94$$



- **Information gain**: expected reduction in entropy when partitioning the training data according to the attribute

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Decision trees: what is the 'best' attribute?

A detour

$S:\{9+,5-\}$

$E = 0.94$

Humidity

high: $S:\{3+,4-\}$, $E = 0.985$

normal: $S:\{6+,1-\}$, $E = 0.592$

$Gain(S,H) = .94 - (7/14).985 - (7/14).592$
$= 0.151$

$S:\{9+,5-\}$

$E = 0.94$

Wind

strong: $S:\{6+,2-\}$, $E = 0.811$

weak: $S:\{3+,3-\}$, $E = 1.00$

$Gain(S,W.) = 0.048$

➔Pick humidity over wind as next attribute

Decision trees: there is more to it

A detour

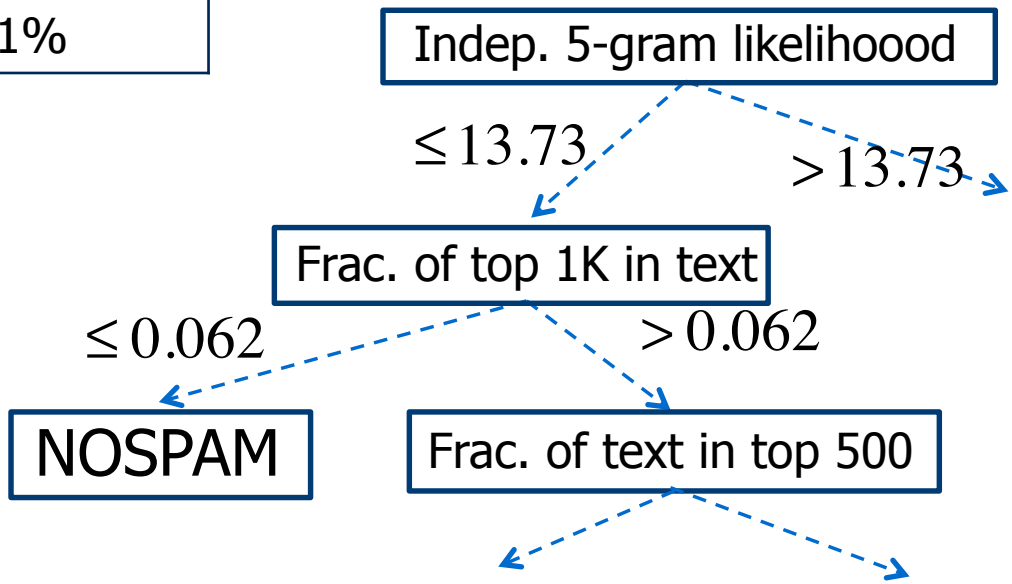
- Algorithm: ID3
  - Improvements exist
  - Pruning, etc.
- Most famous decision tree algorithm: C4.5
- Continuous variables can be included via boolean attributes
  - A_c *is true if* $A<c$, *false otherwise*
- Tom Mitchell: http://www.cs.cmu.edu/~tom/mlbook.html

Classification accuracy

Ntoulas et al., 2006 [9]

- Detection rates

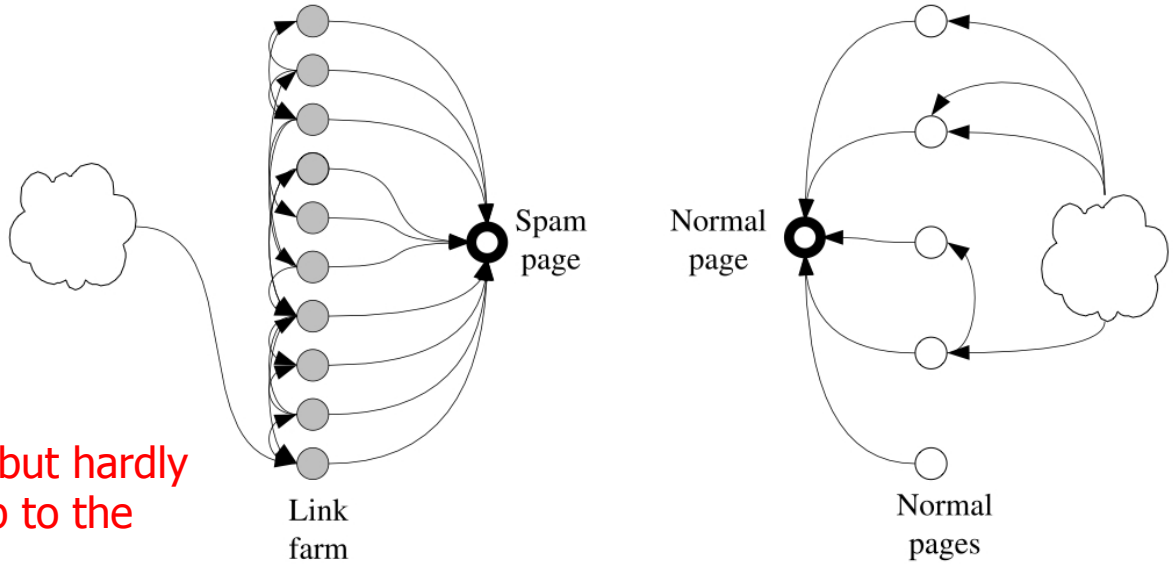
class	recall	precision
SPAM	82%	84%
NOSPAM	97.5%	97.1%



Topological spam

Becchetti et al., 2006 [10]

- Link farm: a densely connected set of pages, created explicitly with the purpose of deceiving a link-based ranking algorithm

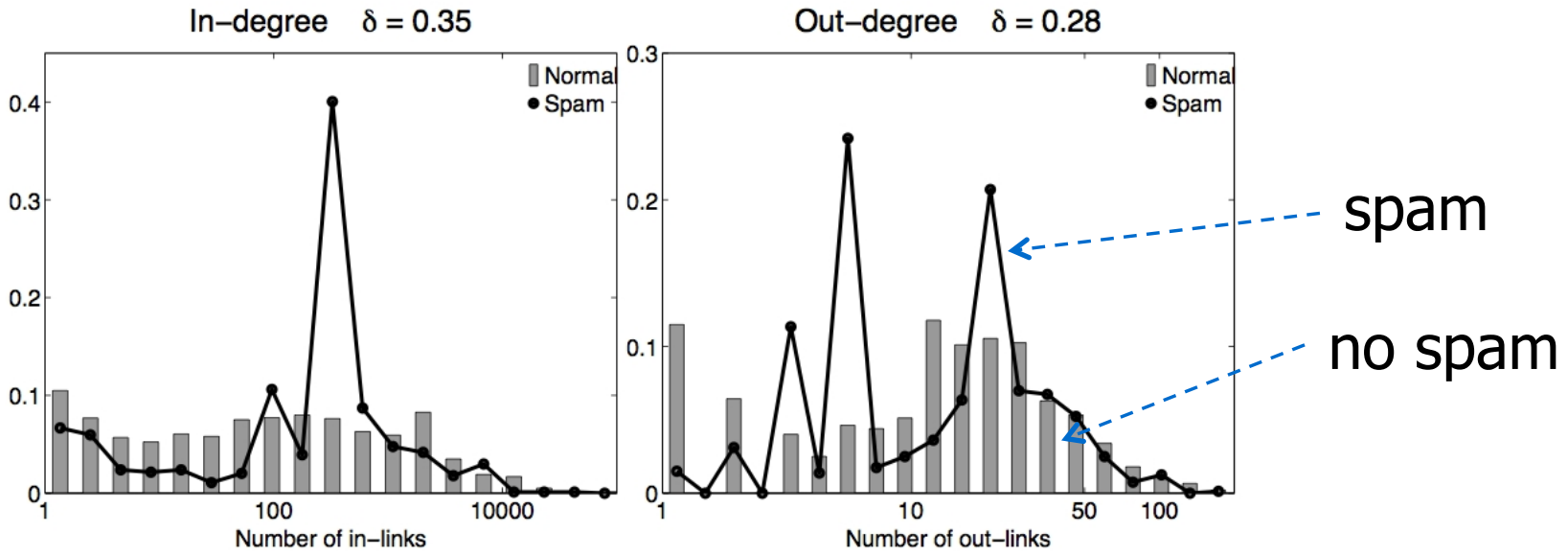


high in-degree but hardly any relationship to the rest of the web

Link based classification features

Becchetti et al., 2006 [10]

- Distribution of in-degree and out-degree

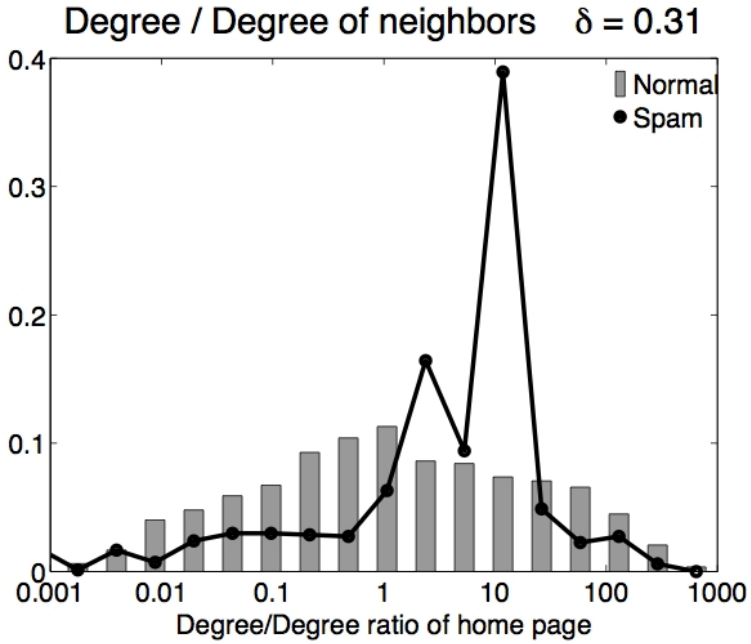


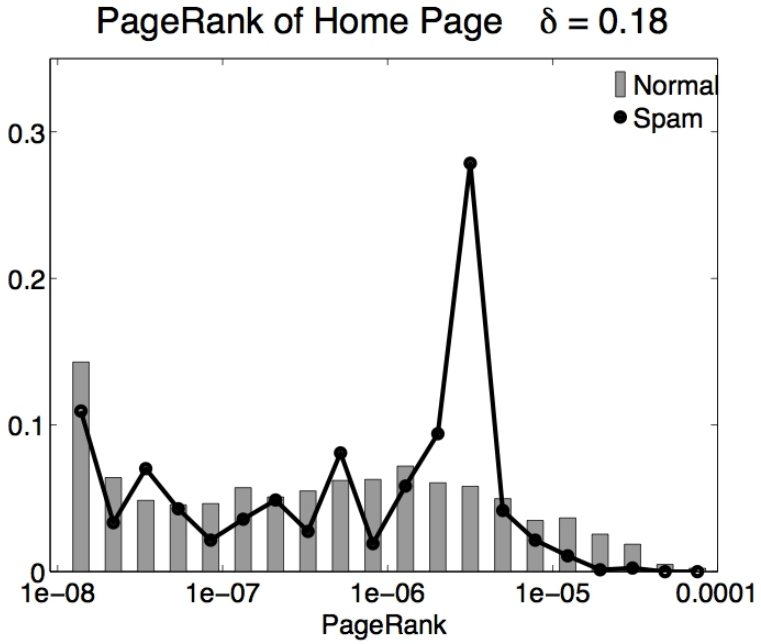
- δ measures the maximum difference of the cumulative distribution functions; the larger the more different the distributions

Link based classification features

Becchetti et al., 2006 [10]

- Degree/degree ratio: ratio between a node's degree and the average degree of its neighbours (in-links and out-links)
- PageRank





Link based classification features

Becchetti et al., 2006 [10]

- **TrustRank**: given a seed of trusted nodes, propagate their scores along the edges
- Intuition: a page with high PageRank but without a relationship with a trusted page is suspicious
- Trusted nodes: pages listed in the Open Directory Project (manually maintained)
- Estimated non-spam mass: amount of PageRank score that a page receives from trusted pages

Link based classification features

Becchetti et al., 2006 [10]

- 163 attributes in total
- Decision tree classifier
- Detection rate: 80%

$$\frac{\#\,spam\ sites\ classified\ as\ spam}{\#\,spam\ sites}$$

- False positive rate: 1-3%

$$\frac{\#\,normal\ sites\ classified\ as\ spam}{\#\,normal\ sites}$$

Ongoing issues in spam detection

- Spam detection is an arms race between the spammer and the search engine
- Unsupervised spam filtering
  - Shown approaches require training data (ham&spam), which is costly and ineffective (spammers learn)
- Unsupervised approaches do not rely on training data
- Image spam (most spam detection methods based on textual content)

Folksonomies

Folksonomies

Delft stadhuis Markt

Delft stadhuis Markt

This photo was taken on May 24, 2011.

71 views 10 comments

This photo belongs to

Gerard Stolk (vers le Midi Carême)'s photos... (7,404)

This photo also appears in

voorjaar 2011 (set)

Tags

Delft • stadhuis • Markt

License

Some rights reserved

Delft stadhuis
Markt

Folksonomies

Delft stadhuis Markt

Delft stadhuis Markt

This photo was taken on May 24, 2011.

71 views 10 comments

This photo belongs to

Gerard Stolk (vers le Midi Carême)'s photos... (7,404)

This photo also appears in

BibSonomy search :: information retrieval search

THE BLUE SOCIAL BOOKMARK AND PUBLICATION SHARING SYSTEM.

EN · DE

username : password sign in

home groups popular sign in

BOOKMARKS (971)

Benutzerorientierte Optimierung von Inform...

3 days and 15 hours ago by vicusnigri

benutzerorientierung filetype:pdf information_retr...

The Shortcomings of Full-Text Searching | m...

3 days and 15 hours ago by vicusnigri

information_retrieval präsentation volltextsuche

Using Sampled Data and Regression to Mer...

7 days ago by estebancacavelo

information merging results retrieval

PUBLICATIONS (12191)

Who should I cite: learning literature search ...

Steven Bethard, and Dan Jurafsky *Proceedings of the 19th ACM...*

21 hours and 17 minutes ago by jaeschke

research recommender paper item citation anal...

Recommending citations for academic papers

Trevor Strohman, W. Bruce Croft, and David Jensen *Proceeding...*

21 hours and 27 minutes ago by jaeschke

research recommender paper item citation anal...

Indexing and retrieval of scientific literature

Steve Lawrence, Kurt Bollacker, and C. Lee Giles *Proceedings o...*

a day and 12 hours ago by jaeschke

science retrieval research publication informatio...

tags

(alpha | freq) (cloud | list) (minfreq 1 | 2 | 5 | 10 | 50)

2.0 book classification conference data datamining dblp finden folksonomies folksonomy imported indexing Information information information+retrieval information-retrieval Information_Retrieval information_retrieval InformationandKnowledgeManagement informationen InformationRetrieval informationretrieval Informationswissenschaft IR ir irhhu knowledge+management lecture metadata mining nlp research Retrieval retrieval science search semantic semantics semanticweb Stock stock suchen system:imported tagging tutorial v0805 Web web web2.0 XML

information merging
results retrieval

Social tagging systems

- What: services allow the labeling of content with keywords (tags) chosen freely by the users
  - **Narrow** tagging rights: tagging resources limited to one/several users (e.g. Flickr)
  - **Broad** tagging rights: tagging of resources by the entire community (e.g. Bibsonomy)
- Why: content collection, content management and sharing
- Alternative to existing top-down categorization techniques (taxonomy, dictionary)

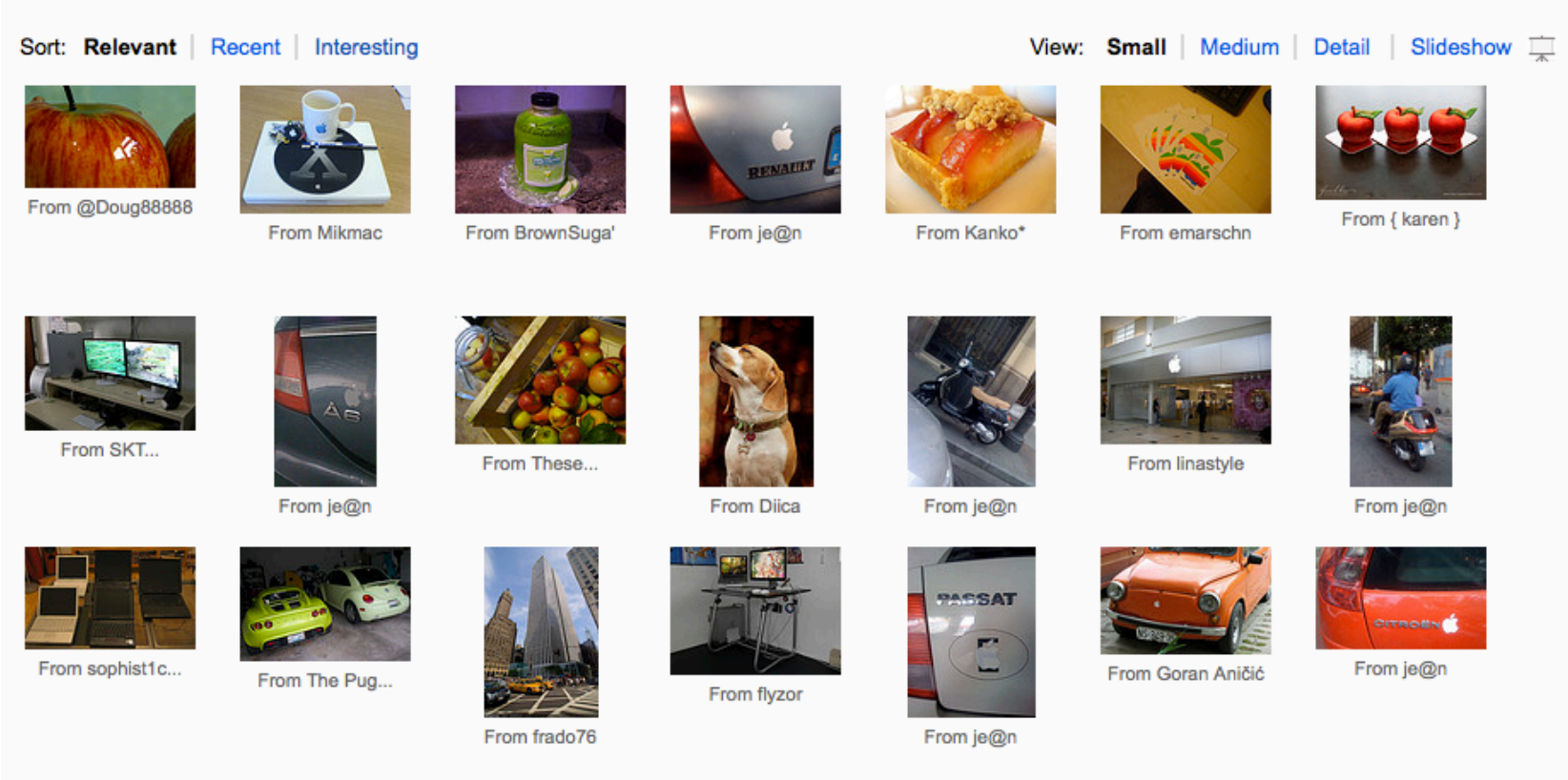
- Folksonomy: social tagging data, social tagging communities

Advantages & disadvantages

- Advantages
  - Easily accessible to users (adding tags is simple)
  - Stay up-to-date as long as user provides tags
  - Items can belong to many "categories"
  - Self-moderation
- Disadvantages
  - Flat structure
  - Uncontrolled vocabulary which leads to ambiguity
  - Lack of precision
  - Noise (spam, malicious users)

Tag ambiguity on Flickr

Searching for "apple" on Flickr (14.03.2012)



Applications of Folksonomies

- Item recommendation
  - Recommend new interesting items to folksonomy users based on their profile
- Tag recommendation
  - Aid user who adds a resource to the system by suggesting potentially matching tags
- Personalized web search

Formally

Vallet et al., 2010 [11]

- Folksonomy F can be defined as a tuple $F=\{T,U,D,A\}$
- $T=\{t_1,\ldots,t_L\}$ is the set of tags expressed by the folksonomy
- $U=\{u_1,\ldots,u_M\}$ is the set of users that annotate documents
- $D=\{d_1,\ldots,d_n\}$ is the set of documents that are annotated with tags T

- The set of annotations of each tag t_l to document d_n by user u_m

$$A = \{(u_m, t_l, d_n)\} \in U \times T \times D$$

User and document profiles

Vallet et al., 2010 [11]

- User profile

$$\vec{u}_m = (u_{m,1},...,u_{m,L}),\ where\ u_{m,l} = |\{(u_m, t_l, d) \in A \mid d \in D\}|$$

Based on the number of times the user has tagged documents with tag t_i

- Document profile

$$\vec{d}_n = (d_{n,1},...,d_{n,L}),\ where\ d_{n,l} = |\{(u, t_l, d_n) \in A \mid u \in U\}|$$

Based on the number of times the document has been tagged with tag t_i

Personalization step

Vallet et al., 2010 [11]

- Given query Q, the top s documents retrieved by a search system are re-ranked according to the user and document profiles

$$tf.if(u_m, d_n) = \sum_l \left(tf_{u_m}(t_l) \times iuf(t_l) \times tf_{d_n}(t_l) \times idf(t_l) \right)$$

user-based tag frequency

user-based inverse tag frequency

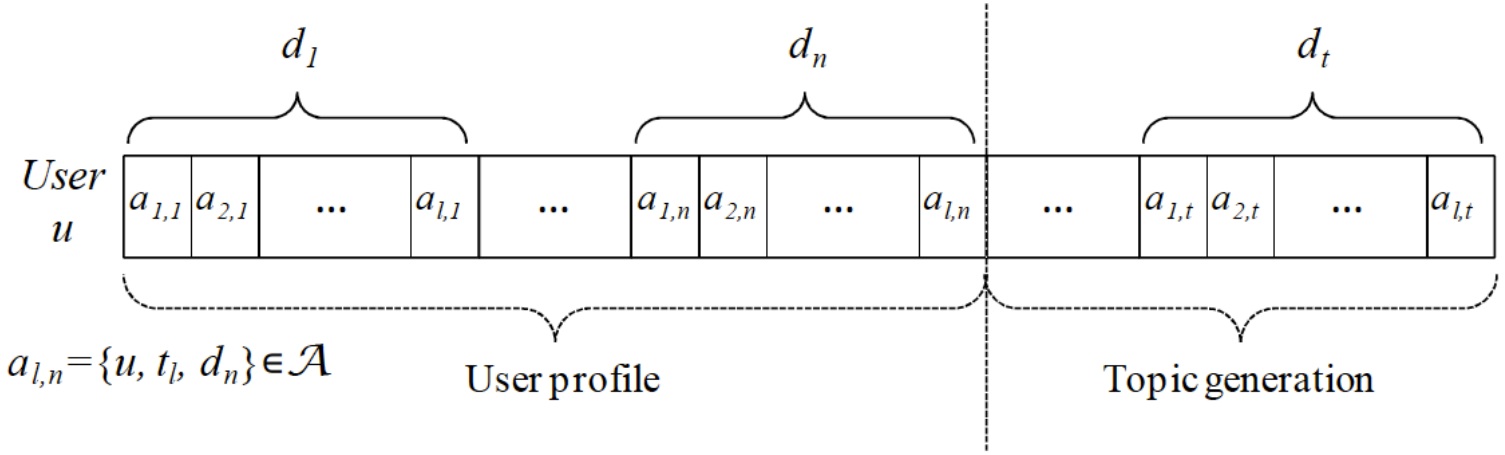
document-based tag frequency

document-based inverse tag frequency

Evaluation

Vallet et al., 2010 [11]

- Document d is relevant to user u if it occurs in his profile
- Tagging information of each user is split into user profile partition and automatic topic generation partition



Evaluation

Vallet et al., 2010 [11]

- Topics are generated from each document in the test partition
  1. Extract the k most popular tags of d
  2. Use extracted tags as query to a Web search engine and return the top R results
  3. If d is not in R, discard the topic
  4. Apply personalization to the result list & report reciprocal rank of d

$$RR = \frac{1}{r}$$

`yahoo.com`	➔{yahoo, search, email, news, searchengine,…}
`stackoverflow.com`	➔{programming, reference, software,…}
`bbc.co.uk`	➔{news, bbc, media, uk, world, daily,…}

Results

Vallet et al., 2010 [11]

- Social tagging sytem: Delicious
- 2000 users: 100 bookmarks per user, 90% for profile creation and 10% for topic generation
- 161542 documents and 69930 distinct tags
- Web search engine: Yahoo!

	Web search	Web search + tf.if
MRR	0.329	**0.402**
P@5	0.452	**0.565**
P@10	0.579	**0.690**
P@20	0.708	**0.798**

What else can be done with folksonomies?

- Rank photos based on their **attractiveness** by exploiting the community feedback on Flickr; learn a classifier based on assigned tags (ugly, beautiful, gross, awesome, …)
- **Locate expert users** in social tagging systems based on the quality of his resources
- **Predict the latitude/longitude of a photo** based on Flickr tags
- …

Summary

- Web structure
- Evolution of the Web
- Personalization on the Web
- Fighting the spammers!

Sources

1) Introduction to Information Retrieval. Manning et al. 2008.
2) Information retrieval. Keith van Rijsbergen, 1979.
3) Managing gigabytes, Witten et al. 1999.
4) Graph structure in the Web. Broder et al. 1999.
5) The PageRank Citation Ranking: Bringing Order to the Web. Page et al. 1999.
6) A taxonomy of Web search. Broder. 2002.
7) The indexable Web is more than 11.5 billion pages. Gulli & Signorini. 2005.
8) A Large-Scale Study of the Evolution of Web Pages. Fetterly et al. 2003.
9) Detecting spam web pages through content analysis. Ntoulas et al. 2006.
10) Link-based characterization and detection of web spam. Becchetti et al. 2006.
11) Personalizing web search with folksonomy-based user and document profiles. Vallet et al. 2010.
12) Authoritative Sources in a Hyperlinked Environment. Kleinberg. 1998.
13) Thumbs up? Sentiment classification using machine learning techniques. Pang et al. 2002.
14) Web spam taxonomy. Gyöngyi et al. 2004
15) Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. Fetterly et al. 2004.
16) Improving cloaking detection using search query popularity and monetizability. Chellapilla et al. 2006